MMCTest – A Safe Algorithm for Implementing Multiple Monte Carlo Tests

Axel Gandy Georg Hahn
Department of Mathematics, Imperial College London

Abstract

We are interested in testing multiple hypotheses using tests that can only be evaluated by simulation such as permutation tests or bootstrap tests. This article introduces a sequential algorithm which modifies standard procedures which work with exact p-values, such as the Benjamini & Hochberg False Discover Rate (FDR) procedure, and controls the number of samples being drawn for each hypothesis. We show that, with arbitrary high probability, the algorithm gives the same classification as the original procedure with the exact p-values. The method is not only applicable to the original FDR procedure but also extends to a wider class of FDR controlling procedures and to controlling the Familywise Error Rate using the Bonferroni correction. At any stage, the algorithm can be interrupted and returns a set of hypotheses which can already be classified with satisfactory precision and a set of hypotheses whose decision is still pending.

Keywords: Bootstrap/resampling, Computationally Intensive Methods, Multiple Comparisons, False Discovery Rate, Sequential Algorithm

1 Introduction

Consider multiple hypotheses to be tested for statistical significance using a procedure which corrects for the multiplicity, such as the method by Benjamini and Hochberg (1995) or the Bonferroni correction (Bonferroni, 1936). Standard procedures require exact knowledge of all p-values. We consider the case where p-values are not known exactly and can only be computed by simulation. For example, this occurs when using bootstrap or permutation tests. Recent studies involving such tests include Pekowska et al. (2010) using data from a genome data archive, Lage-Castellanos et al. (2010) using brain activity data and Jiao and Zhang (2010) using microarray data.

This article introduces `MMCTest`, an algorithm to implement multiple Monte-Carlo tests. The algorithm is sequential: it starts with all hypotheses being unclassified and then proceeds taking samples until all but a certain number of hypotheses have been classified or until a certain effort is reached. The proposed algorithm can be stopped earlier while having the same guarantee on the probability of misclassifications. When being stopped before all hypotheses have been classified, the algorithm returns three sets: the significant, the not significant and the not yet classified hypotheses.

The algorithm gives, with pre-specified probability, the same classification (rejected and non-rejected hypotheses) as the classification based on the exact p-values. It aims to use fewer samples for hypotheses which can already be classified with sufficient confidence and more samples for hypotheses which cannot be classified yet.

Sandve et al. (2011) suggested an algorithm, called `MCFDR`, which is related to our approach. It is a modification of the sequential method by Besag and Clifford (1991). In contrast to our new method, the `MCFDR` method does not give any guarantees on how the results relate to the FDR procedure applied to the exact p-values.

The basic `MMCTest` algorithm will be described in Section 2. Moreover, Section 2 states conditions which need to be satisfied to guarantee the bound on the probability of classification errors and to guarantee the convergence of the algorithm's output of sets of rejected and non-rejected hypotheses to the sets computed using the exact p-values. In Section 3 we show that the multiple testing procedure by Benjamini and Hochberg (1995) and the Bonferroni correction satisfy these conditions. We also discuss the extension of the FDR procedure using estimators by Pounds and Cheng (2006). The dependence of the computational effort of the proposed algorithm on certain parameters is assessed empirically in Section 4. Furthermore, its performance is compared to the `MCFDR` algorithm. Section 5 contains a concluding discussion.

All proofs can be found in the Appendix. The `MMCTest` algorithm is implemented in an R-package (`simctest`, available on CRAN, The Comprehensive R Archive Network).

In this article, $|\cdot|$ denotes the number of elements in a finite set and the length of an interval. Moreover, $\|\cdot\|$ denotes the Euclidian norm of a vector and A^c denotes the complement of $A \subseteq \{1,\dots,m\}$ with respect to $\{1,\dots,m\}$, where m denotes the number of hypotheses under consideration.

2 Description of the algorithm

2.1 Basic algorithm

Consider testing m hypotheses $H_{01},\dots,H_{0m}$ having corresponding test statistics $T_1,\dots,T_m$ and observed values $t_1,\dots,t_m$. A large value of t_i shall indicate evidence against H_{0i}. Moreover, let p_i^* denote the exact p-value belonging to the potentially estimated hypothesis H_{0i}. We assume that $p_1^*,\dots,p_m^*$ are not available analytically, but have to be obtained through simulations.

We assume that for every potentially estimated hypothesis H_{0i}, $i \in \{1,\dots,m\}$, we can obtain samples from the test statistic T_i under the null hypothesis. Thus, by considering if these realizations exceed the observed value t_i, we can generate independent random variables $X_{ij} \sim \text{Bernoulli}(p_i^*)$, $j \in \mathbb{N}$.

Suppose that $h : [0,1]^m \to \mathcal{P}(\{1,\dots,m\})$ takes a vector of p-values and returns the set of indices of hypotheses to be rejected, where $\mathcal{P}$ denotes the power set. We will call any such function a *multiple testing procedure*. Ultimately, we are interested in obtaining $A^* := h(p^*)$, which we refer to as the true set of rejections.

For example, h can be the FDR controlling procedure by Benjamini and Hochberg (1995) with threshold $\alpha > 0$, which works as follows. Given m p-values $p_1,\dots,p_m$, their order statistic is denoted by $p_{(1)} \le p_{(2)} \le \dots \le p_{(m)}$. In case of a tie, equal values are assigned a rank in arbitrary order. Let k be the largest index i for which $p_{(i)} \le \frac{i}{m}\alpha$. Then, rejecting all hypotheses corresponding to $p_{(1)},\dots,p_{(k)}$ controls the False Discovery Rate at threshold α. The procedure can also be expressed as

$$h(p) = \left\{ i \in \{1,\dots,m\} :\ \exists j : r_p(j) \ge r_p(i) \text{ and } m\frac{p_j}{r_p(j)} \le \alpha \right\},$$

where $r_p(i)$ denotes the rank of p_i in the sorted sequence of p-values.

We call a multiple testing procedure h *monotonic* if $h(p) \supseteq h(q)\ \forall p \leq q$, where $p, q \in [0,1]^m$, i.e. if lower p-values lead to more rejections. We show in Theorem 5 in Section 3.1 that the Benjamini-Hochberg procedure is monotonic.

The following generic algorithm is designed for monotonic multiple testing procedures. It iteratively controls the set of hypotheses for which further samples need to be drawn by refining confidence intervals for every p_i^* through Monte-Carlo sampling. At iteration n, the confidence interval for the p-value p_i^* is denoted by I_i^n. For simplicity, we work with closed confidence intervals.

Algorithm 1. ***[MMCTest]***

1. $n := 0$, $\underline{A}_0 := \emptyset$, $\overline{A}_0 := \{1, \ldots, m\}$.
2. *While* $\big(|\overline{A}_n \setminus \underline{A}_n| > c$ *and total number of samples is below an upper limit*$\big)$
   (a) $n := n + 1$.
   (b) *Compute confidence intervals* I_i^n *for all* $i \in \{1, \ldots, m\}$ *by calling a subalgorithm* `ImproveIntervals`$(\overline{A}_n \setminus \underline{A}_n)$.
   (c) *Set* $\underline{A}_n := h((\max I_i^n)_{i=1,\ldots,m})$, $\overline{A}_n := h((\min I_i^n)_{i=1,\ldots,m})$.
3. *Return* $(\underline{A}_n, \overline{A}_n)$.

Remark 1.
1. *The subalgorithm* `ImproveIntervals` *takes a set as argument and computes refined confidence intervals for all indices in this set. A particular implementation is presented in Section 2.2 which ensures that the probability of no false classifications is at least* $1 - \epsilon$, *where* $\epsilon \in (0,1)$ *is a constant to be chosen by the user.*
2. *The algorithm runs until at most* $c \geq 0$ *hypotheses are classified or until the total number of samples drawn reaches a pre-specified upper limit.*

Example 1. *An example run of* `MMCTest` *(with* $c = 0$*) is visualized in Figure 1. It uses* $m = 10$ *hypotheses and the Benjamini-Hochberg FDR controlling procedure with threshold* $\alpha = 0.4$. *The confidence intervals have been computed as described in Section 2.2. Columns show different iterations, the upper row shows the computation of* $\overline{A}_n$, *the lower row shows the computation of* $\underline{A}_n$. *Only the lower (upper) end of the confidence interval matters for the computation of* $\overline{A}_n$ *(*$\underline{A}_n$*) thus the hypotheses are ordered by their lower (upper) confidence limit in the upper (lower) row. In this example this turns out to be the same ordering. In the second iteration (left column),* `MMCTest` *has already classified the last hypothesis as being non-rejected as the lower confidence limit of its p-value lies above the line connecting the points* $(0,0)$ *and* (m, α) *which we call the Benjamini-Hochberg line. All other hypotheses are still undecided and thus their confidence intervals will be refined. After a few additional iterations (middle column), the seven smallest values can be classified as rejected as the upper confidence limit of the seventh value is below the line. Likewise, the confidence interval of the ninth value has now been shrunk to be entirely above the line which classifies this value as non-rejected. The eighth p-value is still unclassified as its confidence interval overlaps with the line. After refining the confidence interval further, the algorithm stops in the situation depicted in the right column: the two classifications in* $\overline{A}_n$ *and* $\underline{A}_n$ *have converged.*

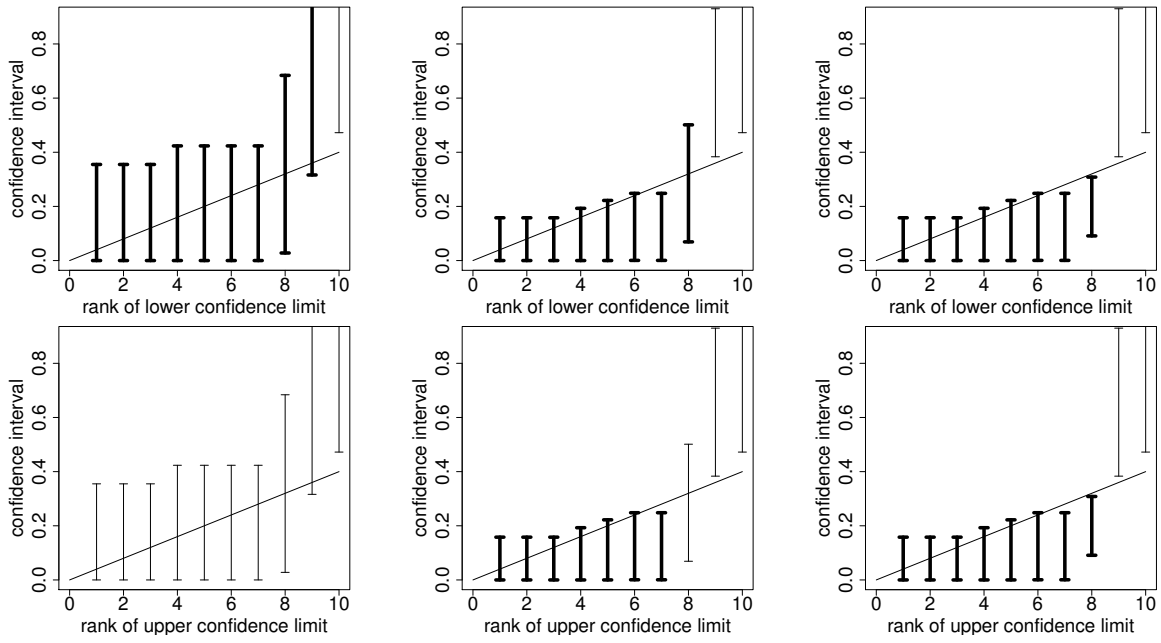

Figure 1: Visualization of `MMCTest` after the second iteration (left), after a few additional iterations (center) and after the last iteration (right); in the upper row, the Benjamini-Hochberg procedure h is applied to lower confidence limits, where the set $\overline{A}_n$ of rejected hypotheses is visualized with bold confidence intervals; the lower row shows the rejections $\underline{A}_n$ when applying h to upper confidence limits.

Under conditions, the monotonicity of h implies immediately that the sequence of sets $\underline{A}_n$ is increasing, that the sequence of sets $\overline{A}_n$ is decreasing and that each $\underline{A}_n$ ($\overline{A}_n$) is a subset (superset) of the true set of rejections A^*.

Lemma 1. *Assume that h is monotonic.*

1. *If all confidence intervals I_i^n are nested, i.e. if $I_i^{n+1} \subseteq I_i^n \ \forall i, n$, then $(\underline{A}_n)_{n\in\mathbb{N}} \nearrow$ and $(\overline{A}_n)_{n\in\mathbb{N}} \searrow$.*
2. *If $p_i^* \in I_i^n \ \forall i, n$, then $\underline{A}_n \subseteq A^* \subseteq \overline{A}_n \ \forall n \in \mathbb{N}$.*

2.2 Computation of confidence intervals

In this section we state conditions on the algorithm `ImproveIntervals` and on the multiple testing procedure h which guarantee that the classification by `MMCTest` is correct with high probability and that all hypotheses will be classified. After that, we present one specific implementation of algorithm `ImproveIntervals` which satisfies these conditions.

The main theorem stated in this section relies on two properties of the multiple testing procedure h, collected in the following condition.

Condition 1. 1. *h is monotonic.*

2. *Let $p, q \in [0,1]^m$. If $q_i \le p_i \ \forall i \in h(p)$ and $q_i \ge p_i \ \forall i \notin h(p)$, then $h(p) = h(q)$.*

Besides asking for monotonicity, Condition 1 ensures that lowering the p-value of a rejected hypothesis or increasing the p-value of a non-rejected hypothesis does not change the result of h.

Recall that for $i \in \{1, \dots, m\}$, independent random variables $X_{ij} \sim \text{Bernoulli}(p_i^*)$, $j \in \mathbb{N}$, simulate test statistics exceeding the null hypothesis. The next condition addresses subalgorithm `ImproveIntervals` and states that the length of all confidence intervals which are not stopped from being improved goes to zero.

Condition 2. *Let $B_n \subseteq \{1, \dots, m\}$, $n \in \mathbb{N}$, and let $X_{ij} \in \{0,1\}$, $j \in \mathbb{N}$, be any sequence for $i \in \{1, \dots, m\}$. Then $|I_i^n| \to 0$ as $n \to \infty$ for all $i \in \limsup_{n\to\infty} B_n = \bigcap_{\nu\ge 1} \bigcup_{n\ge\nu} B_n$, where $\{I_i^n\} =$ `ImproveIntervals`(B_n).*

The main theorem can now be stated.

Theorem 2. *Suppose Conditions 1 and 2 hold and suppose that there exists $\delta > 0$ such that $p \in [0,1]^m$ and $\|p - p^*\| < \delta$ imply $h(p) = h(p^*)$. Then, on the event $\{p_i^* \in I_i^n \ \forall i, n\}$, both sequences $(\underline{A}_n)_{n\in\mathbb{N}}$ and $(\overline{A}_n)_{n\in\mathbb{N}}$ converge to A^*, i.e. there exists $n_0 \in \mathbb{N}$ such that $\underline{A}_n = \overline{A}_n = A^*$ $\forall n \ge n_0$.*

The condition on p^* in Theorem 2 ensures that p^* has a neighborhood on which h is constant.

The next condition ensures that the confidence intervals computed by subalgorithm `ImproveIntervals` have a guaranteed minimal joint coverage probability.

Condition 3. *For a given $\epsilon > 0$, subalgorithm `ImproveIntervals` computes confidence intervals I_i^n in such a way that $\mathbb{P}(p_i^* \in I_i^n \ \forall i, n) \ge 1 - \epsilon$.*

The main theorem and Condition 3 together immediately give a bound on the probability of misclassifications.

Corollary 3. *Under the conditions of Theorem 2 and under Condition 3,*

$$\mathbb{P}(\exists n_0 : \underline{A}_n = A^* = \overline{A}_n \ \forall n \geq n_0) \geq 1 - \epsilon,$$

i.e. the probability that all classifications are correct is at least $1 - \epsilon$.

Next, we give an explicit implementation of the subalgorithm `ImproveIntervals` which will be used in the examples of this article.

Parameters of this algorithm are $\epsilon > 0$, the error probability for any false classification one is willing to tolerate and $(\eta_k)_{k \in \mathbb{N}_0}$, a sequence such that $0 = \eta_0 \leq \eta_1 \leq \ldots$ and $\eta_k \to 1 - (1-\epsilon)^{1/m}$ as $k \to \infty$, which is used to control how ϵ is spent over the iterations of the algorithm. We refer to (η_k) as *spending sequence*. Furthermore, two constants $a \geq 1$ and $\Delta \geq 1$ control how many additional samples are drawn in each iteration. For Example 1 and the examples in Section 4 we have used $a = 1.25$ and $\Delta = 10$.

In the algorithm, two vectors $S, k \in \mathbb{N}_0^m$ keep track of counts. They have to be initialized to $S = k = 0$ before running `MMCTest`.

Algorithm 2. ***[ImproveIntervals(*** B ***)]***

1. $\Delta := a\Delta$.
2. *For all* $i \in B$:
   (a) $S_i := S_i + \sum_{j=k_i+1}^{k_i+\Delta} X_{ij}$.
   (b) $k_i := k_i + \Delta$.
   (c) $I_i^n := \begin{cases} \left[1 - q^{Beta}_{k_i - S_i, S_i+1}(\rho_{k_i}), 1 - q^{Beta}_{k_i+1-S_i, S_i}(1-\rho_{k_i})\right] & 0 < S_i < k_i, \\ \left[0, 1 - \rho_{k_i}^{1/k_i}\right] & S_i = 0, \\ \left[\rho_{k_i}^{1/k_i}, 1\right] & S_i = k_i, \end{cases}$
   where $\rho_{k_i} = (\eta_{k_i} - \eta_{k_i - \Delta})/2$.
3. $I_i^n := I_i^{n-1}$ *for all* $i \notin B$.

Remark 2.
1. *The confidence intervals computed in step 2.(c) of Algorithm 2 are the exact "Clopper-Pearson" confidence intervals, see Clopper and Pearson (1934). The quantiles* $q^{Beta}_{\alpha,\beta}(\epsilon)$ *of the Beta*(α, β) *distribution being used are defined by* $\mathbb{P}(Z \leq q^{Beta}_{\alpha,\beta}(\epsilon)) = \epsilon$ *for a random variable* Z *with probability density function* $\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} z^{\alpha-1}(1-z)^{\beta-1}$. *Computing confidence intervals in this way leads to slightly conservative coverage probabilities in practice, see Li et al. (2009).*
2. *The "Clopper-Pearson" confidence intervals we compute in Algorithm 2 are not necessarily nested and thus the first result of Lemma 1 does not hold. This can easily be fixed by intersecting the current confidence interval with the one computed in the previous iteration.*
3. *With* k_i *being the number of samples drawn up to iteration* n, $1 - \eta_{k_i}$ *is the minimal joint coverage probability of* $I_i^1, \ldots, I_i^n$ *for* p_i^*.

As shown next, Algorithm 2 computes confidence intervals in such a way as to satisfy Condition 2 and Condition 3.

Lemma 4. *Algorithm 2 satisfies Conditions 2 and 3 when using the spending sequence* $\eta_k = \frac{k}{k+r}\left(1-(1-\epsilon)^{1/m}\right)$ *for a constant* $r > 0$*, where* ϵ *is the overall error probability,*

Remark 3. *When being called through* `ImproveIntervals`$(\overline{A}_n \setminus \underline{A}_n)$ *in* `MMCTest`*, Algorithm 2 works as follows: Firstly, the number of additional samples* Δ *drawn in every step is increased geometrically. For all hypotheses which are still considered, i.e. those in* $\overline{A}_n \setminus \underline{A}_n$*, an additional batch of* Δ *samples is drawn. The total number of samples drawn up to iteration* n *for a hypothesis* $i \in \{1,\dots,m\}$ *is stored in* k_i *and the total number of significant test statistics is stored in* S_i*. Then, the exact Clopper-Pearson confidence interval* I_i^n *for* p_i^**,* $i \in \{1,\dots,m\}$*, is computed using* k_i *and* S_i*. The confidence interval remains unchanged for each hypothesis* $i \notin \overline{A}_n \setminus \underline{A}_n$ *which is not considered in the current iteration.*

3 Some properties of multiple testing procedures

In this section we discuss how and under which circumstances several multiple testing procedures, namely the procedure by Benjamini and Hochberg (1995), its extension by Pounds and Cheng (2006) and the Bonferroni (1936) correction, satisfy the conditions of Theorem 2.

3.1 Properties of the Benjamini-Hochberg procedure

The next theorem shows three properties of the Benjamini-Hochberg procedure h given in Section 2.1 which are slightly stronger than Condition 1.

Theorem 5. *1.* h *is monotonic.*

2. Let $p, q \in [0,1]^m$*. If* $q_i \leq \frac{|h(p)|\alpha}{m}\ \forall i \in h(p)$ *and* $q_i = p_i\ \forall i \notin h(p)$*, then* $h(p) = h(q)$*.*

3. Let $p, q \in [0,1]^m$*. If* $q_i = p_i\ \forall i \in h(p)$ *and* $q_i > \frac{\alpha}{m} r_p(i)\ \forall i \notin h(p)$*, then* $h(p) = h(q)$*.*

The second statement shows that all p-values in the set of rejections can be increased up to a certain bound without affecting the result of h. The third statement shows that the result of h is not affected if p-values in the non-rejection area are replaced by arbitrary values above the Benjamini-Hochberg line (see Example 1).

Corollary 6. h *satisfies Condition 1.*

The next lemma states that h is locally constant for almost all arguments:

Lemma 7. *If* $p^* \in [0,1]^m$ *with* $p^*_{(i)} \neq i\alpha/m$*,* $i \in \{1,\dots,m\}$*, then there exists* $\delta > 0$ *such that* $p \in [0,1]^m$ *and* $\|p - p^*\| < \delta$ *imply* $h(p^*) = h(p)$*.*

Lemma 7 thus shows that the condition on p^* in Theorem 2 is satisfied for all p-values except for those lying exactly on the Benjamini-Hochberg line.

3.2 Properties of the Bonferroni correction

The Bonferroni correction controls the Familywise Error Rate, defined by $FWER := \mathbb{P}(V \geq 1)$, where V is the number of hypotheses from the null which have been declared significant (false

positives). The method by Bonferroni (1936) tests all m hypotheses $H_{01}, \ldots, H_{0m}$ at threshold α/m to guarantee FWER $\leq \alpha$. The Bonferroni correction h_B can be stated as

$$h_B(p) = \{i \in \{1, \ldots, m\} : p_i \leq \alpha/m\}.$$

The output $h_B(p)$ is the set of rejected indices.

Similarly to Theorem 5, the following theorem states two key properties of h_B which are slightly stronger than the corresponding statements of Condition 1.

Theorem 8. *1. h_B is monotonic.*

2. Let $p, q \in [0,1]^m$. If $q_i \leq \frac{\alpha}{m}\ \forall i \in h_B(p)$ and $q_i > \frac{\alpha}{m}\ \forall i \notin h_B(p)$, then $h_B(p) = h_B(q)$.

The second statement of Theorem 8 shows that the result of h_B is not affected if p-values in the rejection (non-rejection) area are replaced by arbitrary values below (above) the constant testing threshold α/m.

Corollary 9. *h_B satisfies Condition 1.*

Similarly to Lemma 7, the Bonferroni correction is locally constant for almost all values:

Lemma 10. *For all $p^* \in ([0, \alpha/m) \cup (\alpha/m, 1])^m$ there exists $\delta > 0$ such that $p \in [0,1]^m$ and $\|p - p^*\| < \delta$ imply $h_B(p^*) = h_B(p)$.*

Lemma 10 thus shows that the condition on p^* in Theorem 2 is satisfied for all p-values except for those lying exactly on the threshold α/m.

3.3 Extension to other FDR-controlling procedures

Benjamini and Hochberg (1995) showed that their procedure actually controls the FDR at threshold $\frac{m_0}{m}\alpha \leq \alpha$ instead of α, where m_0 is the unknown number of true null hypotheses. A stronger control of the FDR can thus be archived by using the threshold α/π_0, where the proportion of true null hypotheses $\pi_0 := m_0/m$ has to be estimated. Pounds and Cheng (2006) propose to use the estimator $\tilde{\pi}_0(p) = \min(1, \frac{2}{m}\sum_{i=1}^{m} p_i)$.

In this section, we explicitly regard the Benjamini-Hochberg procedure as being a function of both p-values and threshold α,

$$\begin{aligned} h : [0,1]^m \times [0,1] &\to \mathcal{P}(\{1, \ldots, m\}) \\ (p, \alpha) &\mapsto \left\{ i \in \{1, \ldots, m\} :\ \exists j : r_p(j) \geq r_p(i) \text{ and } m\frac{p_j}{r_p(j)} \leq \alpha \right\}. \end{aligned}$$

Using this notation, the multiple testing procedure by Pounds and Cheng (2006) can be written as $h_{PC}(p) = h(p, \alpha/\tilde{\pi}_0(p))$.

We have already shown in Theorem 5 that h is a decreasing function in p. The following lemma shows that h is also an increasing function in α.

Lemma 11. *Let $p \in [0,1]^m$. Then $0 \leq \alpha_1 \leq \alpha_2 \leq 1$ implies $h(p, \alpha_1) \subseteq h(p, \alpha_2)$.*

This immediately implies that h_{PC} satisfies the first statement of Condition 1:

Corollary 12. *h_{PC} is monotonic.*

However, the second statement of Condition 1 is not satisfied as shown in the following remark.

Remark 4. *Consider $m = 2$ hypotheses. Let $\alpha = 2/5$ and $p = (1/5, 3/5)$. Then $\tilde{\pi}_0(p) = 4/5$ and $h_{PC}(p) = h(p, 1/2) = \{1\}$. Let $q = (0, 3/5)$. Then $q_i \leq p_i \ \forall i \in h_{PC}(p) = \{1\}$ and $q_i = p_i$ $\forall i \notin h_{PC}(p)$. Now $\tilde{\pi}_0(q) = 3/5$, implying $h_{PC}(q) = \{1, 2\} \neq h_{PC}(p)$. Thus the second statement of Condition 1 is not satisfied.*

When using `MMCTest` and `ImproveIntervals` in conjunction with h_{PC}, the following scenario can occur. In every iteration, `MMCTest` applies h_{PC} to upper and lower confidence levels $(\max I_i^n)$ and $(\min I_i^n)$ and thus applies the Benjamini-Hochberg FDR controlling procedure at two different thresholds $\alpha/\tilde{\pi}_0((\max I_i^n)_{i=1,\ldots,m})$ and $\alpha/\tilde{\pi}_0((\min I_i^n)_{i=1,\ldots,m})$. Those thresholds vary in every iteration as estimates of p-values vary. As more p-values are classified and leave the active set $B_n = \overline{A}_n \setminus \underline{A}_n$, the remaining p-values in B_n are less likely to significantly change the two thresholds. A single confidence interval can get trapped between the Benjamini-Hochberg lines corresponding to the two thresholds, and thus will never be classified.

An ad-hoc correction to `MMCTest` is the following procedure. First, algorithm `ImproveIntervals` is applied as usual to all active hypotheses in B_n. Let N denote the total number of new samples drawn during this step. Second, $N/|B_n^c|$ new samples are drawn for each inactive hypothesis in B_n^c, resulting in the same total number N of new samples spent again on all inactive hypotheses in B_n^c. New confidence intervals are then computed for all p-values $p_1^*, \ldots, p_m^*$. This correction is used for the empirical comparison in Section 4.3. As shown in the proof of Lemma 4, the length of the Clopper-Pearson confidence intervals computed in the `ImproveIntervals` algorithm tends to zero as the sample size tends to infinity. Drawing further samples for each hypothesis in every iteration thus implies convergence of the above correction to the classification using exact p-values.

4 Empirical comparison

This section starts with an empirical assessment of the dependence of the computational effort of `MMCTest` on the number of hypotheses m. After that, we illustrate the convergence of the sets $\overline{A}_n$ and $\underline{A}_n$ to A^*. A simulation in Section 4.2 suggests that the effort for classifying most hypotheses is reasonable, but that the effort for a complete classification can be very large. Section 4.3 contains an empirical comparison of `MMCTest` to `MCFDR` by Sandve et al. (2011).

The p-values used in this section are simulated from a mixture distribution consisting of a proportion π_0 (the proportion of true null hypotheses) from a uniform distribution in $[0, 1]$ and a remaining proportion $1 - \pi_0$ of p-values drawn from a Beta$(0.25, 25)$ distribution. This mixture distribution was already used in Sandve et al. (2011).

In all simulations, Algorithm 2 was used to compute confidence intervals in step 2.(b) of `MMCTest`. The batch size Δ in Algorithm 2 was increased geometrically by a factor of $a = 1.25$ in every iteration, starting with $\Delta = 10$. The spending sequence was $\eta_k = \frac{k}{k+r}\left(1 - (1-\epsilon)^{1/m}\right)$ as given in Lemma 4 with $\epsilon = 0.01$ and $r = 1000$. Multiple testing was always performed at threshold $\alpha = 0.1$. The parameter π_0 will be given for every simulation. In the entire section, the Benjamini-Hochberg procedure h as defined in Section 2.1 has been used to control the FDR. The `MCFDR` procedure has one tuning parameter, which we set to the recommended value $(h = 20)$.

The effort of `MMCTest` is measured in terms of N, the total number of samples drawn during a run.

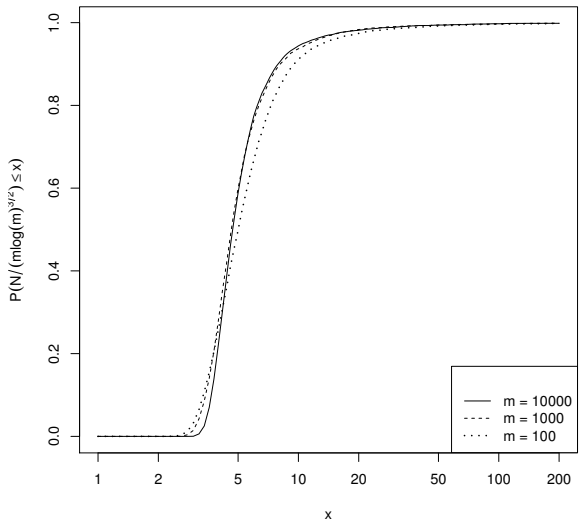

Figure 2: Empirical CDF of $N/(m\log(m)^{1.5})$ for $c = 3$ and $m \in \{100, 1000, 10000\}$, where N denotes the total number of samples.

4.1 Dependence of the effort on the number of hypotheses

How does the number of samples N depend on the number of hypotheses? In the following simulation, the distribution of N turns out to be roughly proportional to $m\log(m)^{3/2}$.

Figure 2 shows the empirical CDF of $N/(m\log(m)^{3/2})$ for $m \in \{100, 1000, 10000\}$. Each empirical CDF was computed based on 10000 runs. For each value of m, the algorithm was run until all but $c = 3$ hypotheses were classified. P-values were uniformly distributed in $[0, 1]$ ($\pi_0 = 1$). As the empirical CDFs match fairly well, Figure 2 indicates that the dependence of N on the number of hypotheses m is roughly of the order $m\log(m)^{3/2}$.

4.2 Dependence of the effort on the stopping rule

Figure 3 illustrates the convergence of the two sets $\underline{A}_n$ and $\overline{A}_n$ to A^* in a single run of `MMCTest`, which we have shown formally in Theorem 2. P-values were generated from the mixture distribution described at the beginning of Section 4 with $\pi_0 = 0.9$ and `MMCTest` was run until all but $c = 10$ hypotheses were classified.

As most null hypotheses are true ($\pi_0 = 0.9$) we expect a large proportion of hypotheses to be easily classified to lie above the Benjamini-Hochberg line. Indeed, the size of $\overline{A}_n$ drops quickly during the first iterations.

We use a relatively low threshold of $\alpha = 0.1$ and only 10% of the hypotheses come from the alternative. This is why a considerable effort is needed to shrink the confidence intervals sufficiently for them to lie entirely below the Benjamini-Hochberg line. This becomes visible in Figure 3 as the size of $\underline{A}_n$ remains unchanged over a large period of time and increases only at a later stage.

Figure 4 shows the dependence of the effort on the number c of undecided hypotheses. To generate this figure, `MMCTest` has been applied 1000 times in the following way to $m = 5000$

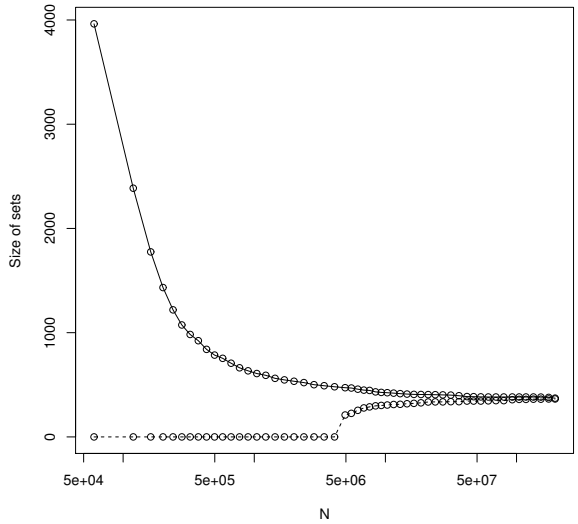

Figure 3: Convergence of $\underline{A}_n$ (dashed) and $\overline{A}_n$ (solid) to A^* for a sample run of `MMCTest`. Size of $\underline{A}_n$ and $\overline{A}_n$ as a function of the number of samples N drawn. The end of every iteration is indicated by a circle. Note the log-scale on the horizontal axis.

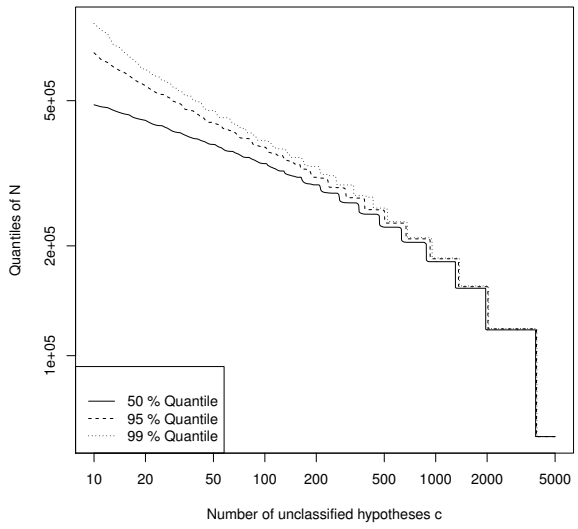

Figure 4: 0.5-, 0.95- and 0.99-quantile of the effort N against the number of unclassified hypotheses c based on 1000 simulations. Log-scale on both axes.

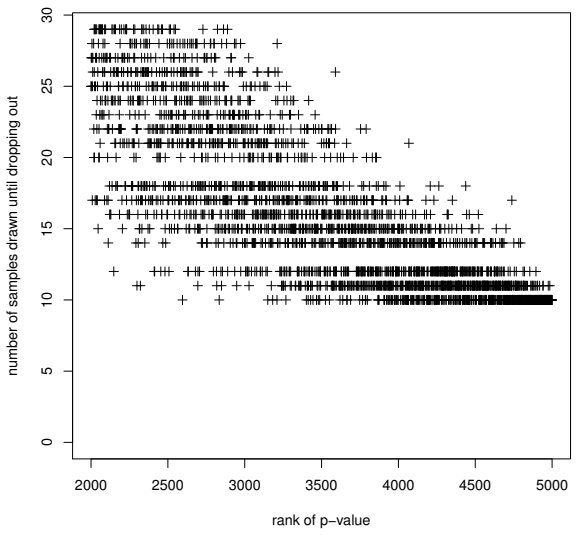

Figure 5: Number of samples drawn until leaving the active set $B_n = \overline{A}_n \setminus \underline{A}_n$ for each hypothesis with a p-value of rank 2000 to 5000. P-values were uniformly distributed in $[0, 1]$.

uniformly distributed p-values in $[0, 1]$: The current size c of set $\overline{A}_n \setminus \underline{A}_n$ and the current total number of samples N_c were recorded in each iteration. If several p-values were classified together in an iteration, some c did not have a corresponding N_c. To be conservative, a missing value N_c is set to $N_{c'}$ for the largest $c' < c$ for which $N_{c'}$ is not missing. Each time the algorithm has been run until all but $c = 10$ hypotheses have been classified.

The effort is reasonable for classifying all but a few hypotheses. Classifying the last few hypotheses seems to be computationally intensive.

The steps in Figure 4 are caused by several hypotheses with p-values far off the Benjamini-Hochberg line being classified together. This effect is explored in Figure 5, which is based on a single run with $m = 5000$ uniformly distributed p-values in $[0, 1]$. A cross marks the number of samples drawn until each hypothesis leaves the active set $B_n = \overline{A}_n \setminus \underline{A}_n$. In every iteration n only one additional sample was drawn for each hypothesis in B_n, i.e. we used $\Delta = 1$ and $a = 1$. `MMCTest` has been run until all but 2000 hypotheses were classified. Each horizontal line in the plot shows that several p-values with different ranks leave the active set at the same time. Many hypotheses can be classified after only a few samples.

4.3 Comparison to `MCFDR`

We now compare `MMCTest` to `MCFDR` by Sandve et al. (2011). Both algorithms were applied to the mixture distribution described at the beginning of Section 4 using proportions π_0 of true null hypotheses ranging from 0 to 1 in steps of 0.01.

`MCFDR` uses h_{PC} (see Section 3.3) by Pounds and Cheng (2006) to control the FDR. Thus h_{PC} is used for `MMCTest` as well, using the procedure described at the end of Section 3.3.

Figure 6(a) shows the number of misclassifications for `MCFDR` in a single run for every value of π_0, obtained by comparing the result of h_{PC} for estimated and exact p-values. Up to 150 hypotheses are misclassified. The corresponding number of samples drawn by `MCFDR` is displayed

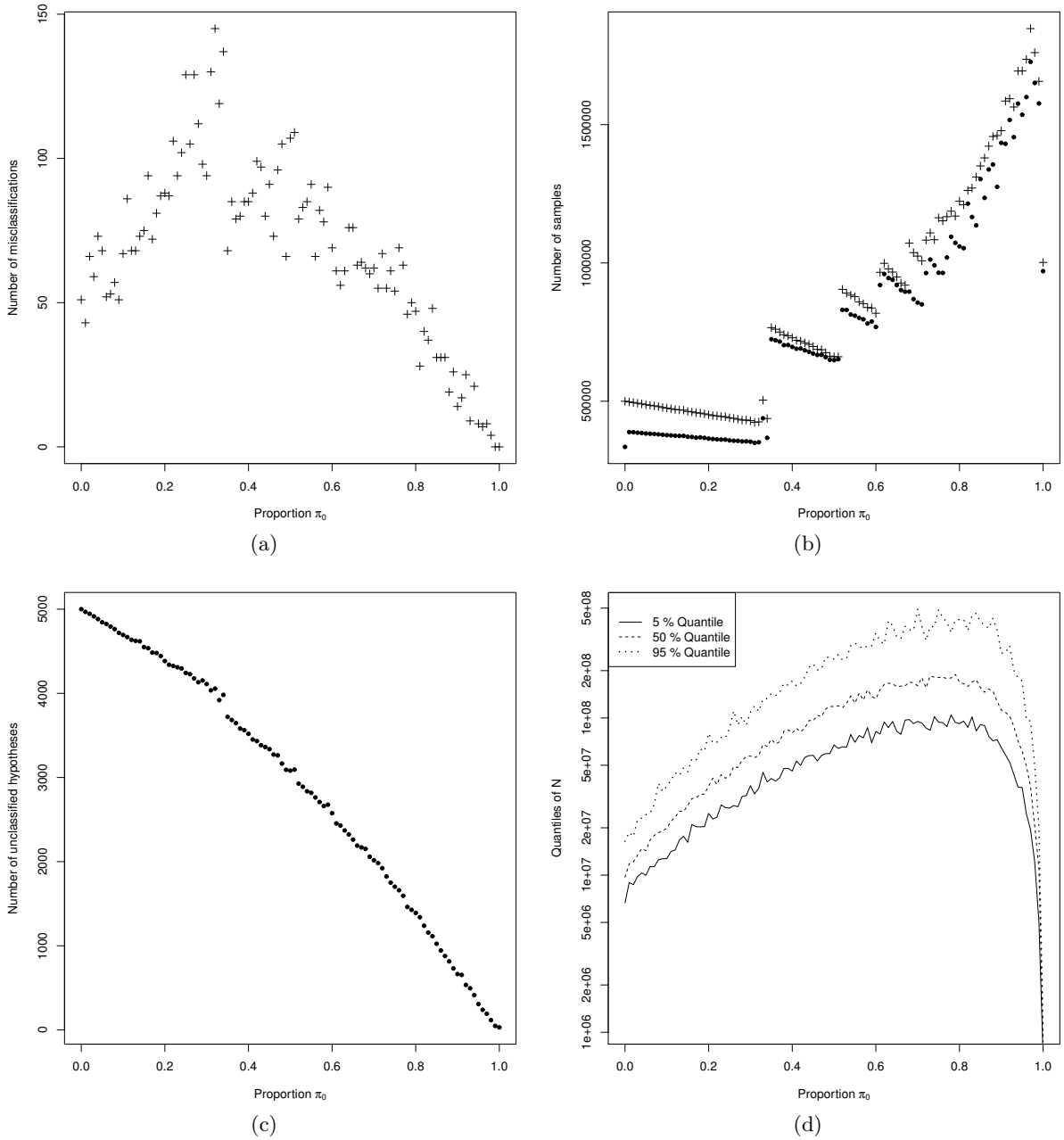

Figure 6: (a) Number of misclassifications for `MCFDR` based on a single run with 5000 hypotheses; (b) Number of samples used by `MCFDR` (crosses) and matched effort for `MMCTest` (dots); (c) Number of unclassified hypotheses for `MMCTest` when restricting effort as in (b); (d) Quantiles of the effort N of `MMCTest` when classifying all but 20 hypotheses (based on 100 replications). Log-scale on the vertical axis.

in plot (b).

First, we restricted the number of samples drawn by `MMCTest` to the number drawn by `MCFDR`, see Figure 6(b), and applied `MMCTest` to the same p-values used to evaluate `MCFDR` in Figure 6(a). Plot (c) then displays the resulting number of unclassified hypotheses by `MMCTest`. For $\pi_0 = 0$ almost all hypotheses are still undecided. This then declines until for $\pi_0 = 1$ almost all hypotheses are classified.

Next, we let `MMCTest` run until all but $c = 20$ hypotheses are classified. Figure 6(d) shows 5%, 50% and 95% quantiles of the number of samples drawn by `MMCTest` based on 100 independent replications of drawing the p-values and running `MMCTest`. The number of samples needed to achieve this classification is mostly substantially higher than the number of samples needed by `MCFDR`. The main advantage of `MMCTest` is that its result is guaranteed to have with high probability no misclassifications as opposed to the large number of misclassifications than can occur when using `MCFDR`.

Suppose we stopped the `MMCTest` procedure before the number of samples used by `MCFDR` was reached. Then, as mentioned before, many hypotheses will not be classified. An ad-hoc classification can be obtained by applying h_{PC} to estimates $\hat{p}_i = S_i/k_i$ of the p-values (see Algorithm 2). This approach leads to a similar number of misclassifications as the `MCFDR` procedure. This illustrates that this variant of our procedure is similar in performance to the `MCFDR` procedure. We do not really recommend this variant; we recommend that whenever the algorithm is stopped one is only using the partioning of the hypotheses into rejected, not rejected and not classified hypotheses as result of the algorithm.

5 Discussion

We presented an open-ended sequential algorithm designed to implement multiple Monte-Carlo tests in the setting where p-values can only be approximated through simulation.

The main feature of the `MMCTest` algorithm is that its output is guaranteed to be correct (with a pre-specified probability), meaning that all classifications are identical to the classifications based on the exact p-values.

Our simulation study shows that a complete classification can be computationally expensive, but that most hypotheses can be classified using a reasonable effort. A detailed theoretical analysis of the computational effort of the proposed algorithm is outside the scope of the present article.

This article identified conditions which guarantee the bounded risk of classification errors and the convergence of the algorithm's output to the classification computed with exact p-values. By verifying those conditions, we showed that the algorithm works for the procedure by Benjamini and Hochberg (1995) as well as for the Bonferroni correction (Bonferroni, 1936) and that it extends to a wider class of FDR controlling procedures. We conjecture that the `MMCTest` algorithm also works for other FDR controlling procedures and for procedures controlling FDR-related criteria (e.g. the False Non-Discovery Rate FNR).

References

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B (Methodological)*, 57(1):289–300.

Besag, J. and Clifford, P. (1991). Sequential Monte Carlo p-values. *Biometrika*, 78(2):301–304.

Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilità. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8:3–62.

Clopper, C. and Pearson, E. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30.

Jiao, S. and Zhang, S. (2010). A mixture model based approach for estimating the FDR in replicated microarray data. *Journal of Biomedical Science and Engineering*, 20(11):317–321.

Lage-Castellanos, A., Martínez-Montes, E., Hernández-Cabrera, J., and Galán, L. (2010). False discovery rate and permutation test: An evaluation in ERP data analysis. *Statistics in Medicine*, 29:63–74.

Li, J., Tai, B., and Nott, D. (2009). Confidence interval for the bootstrap P-value and sample size calculation of the bootstrap test. *Journal of Nonparametric Statistics*, 21(5):649–661.

Pekowska, A., Benoukraf, T., Ferrier, P., and Spicuglia, S. (2010). A unique H3K4me2 profile marks tissue-specific gene regulation. *Genome Research*, 20(11):1493–1502.

Pounds, S. and Cheng, C. (2006). Robust estimation of the false discovery rate. *Bioinformatics*, 22(16):1979–1987.

Sandve, G., Ferkingstad, E., and Nygard, S. (2011). Sequential Monte Carlo multiple testing. *Bioinformatics*, 27(23):3235–3241.

A Appendix

A.1 Proofs of Section 2

Proof of Lemma 1. 1. As all confidence intervals I_i^n are nested by assumption, the sequence $(\max(I_i^n))_{i=1,\dots,m}$ is decreasing. Thus by monotonicity of h,

$$\underline{A}_n = h((\max(I_i^n))_{i=1,\dots,m}) \subseteq h((\max(I_i^{n+1}))_{i=1,\dots,m}) = \underline{A}_{n+1}.$$

Similarly, $\overline{A}_n \supseteq \overline{A}_{n+1}$ as $(\min(I_i^n))_{i=1,\dots,m}$ is increasing.

2. Given that $p_i \in I_i^n \ \forall i, n$, the true p-values satisfy $p_i \leq \max(I_i^n) \ \forall i, n$. When applied to the vectors $(p_i)_{i=1,\dots,m} \leq (\max(I_i^n))_{i=1,\dots,m}$, the monotonicity of h yields

$$A^* = h((p_i)_{i=1,\dots,m}) \supseteq h((\max(I_i^n))_{i=1,\dots,m}) = \underline{A}_n.$$

Similarly, $A^* \subseteq \overline{A}_n$ as $(p_i)_{i=1,\dots,m} \geq (\min(I_i^n))_{i=1,\dots,m}$. □

Proof of Theorem 2. We show that there exists $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$,

$$h((\min I_i^n)_{i\in\{1,\dots,m\}}) = h(p^*) = h((\max I_i^n)_{i\in\{1,\dots,m\}}).$$

To do this, we show $h(p^{(j)}) = h(p^{(j+1)})$, $j \in \{1, \dots, 6\}$, where

$$p^{(1)} := (\min I_i^n)_{i\in\{1,\dots,m\}}, \qquad p^{(4)} := p^*,$$
$$p^{(2)} := \begin{cases} p_i^* & i \in B_n, \\ \min I_i^n & i \notin B_n, \end{cases} \qquad p^{(5)} := \begin{cases} \max I_i^n & i \in B_n, \\ p_i^* & i \notin B_n, \end{cases}$$
$$p^{(3)} := \begin{cases} p_i^* & i \in \overline{A}_n, \\ \min I_i^n & i \notin \overline{A}_n, \end{cases} \qquad p^{(6)} := \begin{cases} \max I_i^n & i \in \overline{A}_n, \\ p_i^* & i \notin \overline{A}_n, \end{cases}$$

and $p^{(7)} := (\max I_i^n)_{i\in\{1,\dots,m\}}$ with $B_n = \overline{A}_n \setminus \underline{A}_n$.

(1) Suppose $\exists i \in \limsup_{n\to\infty} B_n$. By Condition 2, $|I_i^n| \to 0$ as $n \to \infty$. Let δ be as given in the theorem. As $B_n \subseteq \{1,\dots,m\}$ is finite $\forall n \in \mathbb{N}$, there exists $n_0 \in \mathbb{N}$ such that $|I_i^n| < \delta$ for $n \geq n_0$ and all $i \in \limsup_{n\to\infty} B_n$. Hence, $\overline{A}_n = h(p^{(1)}) = h(p^{(2)})$ $\forall n \geq n_0$.

(2) As $p^{(3)} \leq (\max I_i^n)_{i\in\{1,\dots,m\}}$ and as h is monotonic, $\underline{A}_n \subseteq h(p^{(3)})$. As $p_j^{(2)} = \min I_j^n \leq p_j^* = p_j^{(3)}$ $\forall j \in \underline{A}_n$ and $p_j^{(2)} = p_j^{(3)}$ $\forall j \notin \underline{A}_n$, the second statement of Condition 1 yields $h(p^{(2)}) = h(p^{(3)})$ $\forall n \geq n_0$.

(3) As $(\min I_i^n)_{i\in\{1,\dots,m\}} \leq p^{(3)}$ and as h is monotonic, $\overline{A}_n \supseteq h(p^{(3)})$. As $p_j^{(4)} = p_j^* \geq \min I_j^n = p_j^{(3)}$ $\forall j \notin \overline{A}_n$ and $p_j^{(4)} = p_j^{(3)}$ $\forall j \in \overline{A}_n$, the second statement of Condition 1 yields $h(p^{(3)}) = h(p^{(4)}) = h(p^*)$ $\forall n \geq n_0$.

Arguing similarly to (1), (2), (3) we can show $h(p^{(4)}) = h(p^{(5)})$, $h(p^{(5)}) = h(p^{(6)})$ and $h(p^{(6)}) = h(p^{(7)}) = \underline{A}_n$. □

Proof of Corollary 3. By Theorem 2 we have $\underline{A}_n \to A^*$, $\overline{A}_n \to A^*$ as $n \to \infty$ conditional on $\{p_i^* \in I_i^n \ \forall i, n\}$. Under Condition 3, this event occurs with probability $\mathbb{P}(p_i^* \in I_i^n \ \forall i, n) \geq 1 - \epsilon$, hence $\mathbb{P}(\underline{A}_n \to A^*, \overline{A}_n \to A^*) \geq 1 - \epsilon$. □

Proof of Lemma 4. First, we consider an individual Clopper-Pearson confidence interval I_i^n computed in step 2.(c) of Algorithm 2. To ease notation, we drop the indices i and n.

We show that $|I| \leq 2\xi$, where $\xi = \sqrt{\frac{-1}{2k}\log\rho}$ and $\rho = (\eta_k - \eta_{k-\Delta})/2$. The following probabilities are conditional on S and k.

Suppose $S < k$. Then the upper limit p_u of the interval I is the solution to $\mathbb{P}(N \leq S | p = p_u) = \rho$, where $N \sim \text{Binomial}(k, p)$. If $p > S/k + \xi$ then, by Hoeffding's inequality (Hoeffding, 1963),

$$\mathbb{P}(N \leq S) = \mathbb{P}\left(\frac{N}{k} - \mathbb{E}\left(\frac{N}{k}\right) \leq \frac{S}{k} - \mathbb{E}\left(\frac{N}{k}\right)\right) \leq \exp\left(-\frac{2(S/k - p)^2 k^2}{k}\right) < \rho.$$

Thus $p_u \leq S/k + \xi$. If $S = k$ then $p_u = 1$, implying $p_u \leq S/k + \xi$.

Similarly, it can be shown that the lower limit p_l of I satisfies $p_l \geq S/k - \xi$. Hence, $|I| = p_u - p_l \leq 2\xi$.

Now let $(B_n)_{n\in\mathbb{N}}$ and $X_{ij} \in \{0, 1\}$, $j \in \mathbb{N}$, be as given in Condition 2 and suppose $i \in \limsup_{n\to\infty} B_n$. Then $n \to \infty$ implies $k_i \to \infty$ by the construction of the algorithm. Furthermore, the sequence $\eta_k = \frac{k}{k+r}\left(1 - (1-\epsilon)^{1/m}\right)$ satisfies $\eta_k - \eta_{k-1} \sim k^{-2}$, implying $\log(\eta_k - \eta_{k-\Delta}) = o(k)$. As $|I_i^n| \leq 2\sqrt{\frac{-1}{2k_i}\log(\frac{1}{2}(\eta_{k_i} - \eta_{k_i-\Delta}))}$ this implies $|I_i^n| \to 0$ as $n \to \infty$. This proves Condition 2.

Second, we show that Algorithm 2 computes confidence intervals in such a way that $\mathbb{P}(p_i^* \in I_i^n \ \forall i, n) \geq 1 - \epsilon$ and thus satisfies Condition 3.

Let k_i^n denote the value of k_i in iteration n, where $k_i^0 = 0$, and let Δ^n denote the value of Δ in iteration n. Algorithm 2 computes Clopper-Pearson confidence intervals I_i^n such that $\mathbb{P}(p_i^* \notin I_i^n) \leq \eta_{k_i^n} - \eta_{k_i^n - \Delta^n}$. This then yields for $i \in \{1, \dots, m\}$:

$$\begin{aligned}\mathbb{P}(p_i^* \in I_i^n \ \forall n) &= 1 - \mathbb{P}(\cup_n \{p_i^* \notin I_i^n\}) \geq 1 - \sum_{n=1}^{\infty} \mathbb{P}(p_i^* \notin I_i^n) \\ &\geq 1 - \sum_{n=1}^{\infty} (\eta_{k_i^n} - \eta_{k_i^n - \Delta^n}) = (1-\epsilon)^{1/m}.\end{aligned}$$

As (X_{ij}) are independent for $i \in \{1, \dots, m\}$ and $j \in \mathbb{N}$, the joint probability is $\mathbb{P}(p_i^* \in I_i^n \ \forall i, n) = \prod_{i=1}^m \mathbb{P}(p_i^* \in I_i^n \ \forall n) \geq 1 - \epsilon$. Condition 3 is thus satisfied. □

A.2 Proofs of Section 3.1

Proof of Theorem 5. As h is invariant to permutations, we may assume $p_1 \leq \dots \leq p_m$.

1. Let $p \in [0,1]^m$ and $i \in \{1, \dots, m\}$. It suffices to show that $h(p) \supseteq h(q)$ for any $q \in [0,1]^m$ given by $q_j = p_j \ \forall j \neq i$ and $q_i > p_i$.

Let $k := |h(p)|$ be the largest index which is rejected when the Benjamini-Hochberg procedure is applied to p. We need to show that $j \notin h(q) \ \forall j \geq k+1$.

Case 1: $r_q(i) \leq k$. This implies $r_q(j) = j \ \forall j \geq k+1$ and hence $q_j = p_j > \frac{\alpha j}{m} = \frac{\alpha r_q(j)}{m}$. Therefore, $j \notin h(q) \ \forall j \geq k+1$.

Case 2: $r_q(i) \geq k+1$. Let $j \geq k+1$, $j \neq i$. Then the rank of the jth p-value can only drop by one when p_i is replaced by q_i, i.e. $r_q(j) \in \{j-1, j\}$. Thus $q_j = p_j > \frac{\alpha j}{m} \geq \frac{\alpha r_q(j)}{m}$. Furthermore, as $r_q(i) \geq k+1$, q_i takes the position of the former $p_{r_q(i)}$ in the ordered sequence of values from q, i.e. $q_i \geq p_{r_q(i)}$. Hence, $r_q(i) \notin h(p)$ because of $r_q(i) \geq k+1$ and thus $q_i \geq p_{r_q(i)} > \frac{\alpha r_q(i)}{m}$. Therefore, $\{k+1, \dots, m\} \cup \{i\} \notin h(q)$. This proves statement 1.

2. All $i \notin h(p)$ satisfy $p_i > \frac{|h(p)|\alpha}{m}$ whereas by assumption, $q_i \leq \frac{|h(p)|\alpha}{m} \ \forall i \in h(p)$. Hence, using $q_i = p_i \ \forall i \notin h(p)$, it follows that $r_q(i) = r_p(i) \ \forall i \notin h(p)$. Thus, $q_i = p_i > \frac{r_p(i)\alpha}{m} = \frac{r_q(i)\alpha}{m}$ for all $i \notin h(p)$. Hence $h(p)^c \subseteq h(q)^c$.

Conversely, define $\tilde{q} := \max\{q_i : i \in h(p)\}$. As $\tilde{q} \leq \frac{|h(p)|\alpha}{m} < q_i$ for all $i \notin h(p)$ and as there are exactly $|h(p)|$ values $q_i \leq \tilde{q}$, the rank of $\tilde{q}$ in q is precisely $|h(p)|$. As $q_i \leq \tilde{q} \leq \frac{|h(p)|\alpha}{m}$ $\forall i \in h(p)$, all $\{q_i\}_{i \in h(p)}$ are rejected and $h(p) \subseteq h(q)$. This proves statement 2.

3. As $q_i = p_i$ for all $i \in h(p)$, have $h(p) \subseteq h(q)$.

Let $i \notin h(p)$. If $r_q(i) \leq r_p(i)$, then $q_i > \frac{\alpha}{m} r_p(i) \geq \frac{\alpha}{m} r_q(i)$. If $r_q(i) > r_p(i)$, q_i replaces a $q_j > \frac{\alpha}{m} r_p(j)$ at rank $r_p(j)$ in the sorted sequence of q, hence $r_q(i) = r_p(j)$ and $q_i \geq q_j > \frac{\alpha}{m} r_p(j) = \frac{\alpha}{m} r_q(i)$. Thus $q_i > \frac{\alpha}{m} r_q(i) \ \forall i \notin h(p)$, which implies $h(p)^c \subseteq h(q)^c$. This proves statement 3. □

Proof of Corollary 6. The first statement of Condition 1 is satisfied as h is monotonic by Theorem 5.

To prove that the Benjamini-Hochberg procedure h also satisfies the second statement of Condition 1, it suffices to show that for $p, q \in [0,1]^m$, both $q_i \leq p_i \ \forall i \in h(p)$ and $q_i = p_i$ $\forall i \notin h(p)$ as well as $q_i = p_i \ \forall i \in h(p)$ and $q_i \geq p_i \ \forall i \notin h(p)$ imply $h(p) = h(q)$.

Indeed, let $p, q \in [0,1]^m$ be such that $q_i \leq p_i \ \forall i \in h(p)$ and $q_i = p_i \ \forall i \notin h(p)$. We have $p_i \leq \frac{|h(p)|\alpha}{m} \ \forall i \in h(p)$ by definition of h, thus $q_i \leq p_i \leq \frac{|h(p)|\alpha}{m} \ \forall i \in h(p)$ and $h(p) = h(q)$ by statement 2 of Theorem 5.

Similarly, let $p, q \in [0,1]^m$ be such that $q_i = p_i \ \forall i \in h(p)$ and $q_i \geq p_i \ \forall i \notin h(p)$. Using that $p_i > \frac{\alpha}{m} r_p(i) \ \forall i \notin h(p)$ it immediately follows that $q_i \geq p_i > \frac{\alpha}{m} r_p(i) \ \forall i \notin h(p)$ and thus $h(p) = h(q)$ by statement 3 of Theorem 5. □

Proof of Lemma 7. The result of h remains unchanged if all p-values do not change their rank outside of a tie and if no p-value crosses the Benjamini-Hochberg threshold line.

As h is invariant to permutations, we may assume $p_1^* \leq \cdots \leq p_m^*$.
Let $\delta := \min\left(\left\{\frac{p_i^* - p_{i-1}^*}{2} : \ i = 2, \ldots, m \text{ with } p_{i-1}^* < p_i^*\right\} \cup \{|p_i^* - i\alpha/m| : \ i = 1, \ldots, m\}\right)$.

Let $p \in [0,1]^m$ with $\|p - p^*\| < \delta$. Then $p_{i-1}^* < p_i^*$ implies $p_{i-1} < p_{i-1}^* + \delta \leq p_i^* - \delta < p_i$. Thus, by possibly permuting indices corresponding to tied values in p, we may assume $p_1^* \leq \cdots \leq p_m^*$ and $p_1 \leq \cdots \leq p_m$. The ranks of the p-values in p^* and p are therefore the same.

Furthermore, $|p_i - p_i^*| < \delta \leq |p_i^* - i\alpha/m|$ for all $i \in \{1, \ldots, m\}$, implying that p_i^* and p_i lie on the same side of the Benjamini-Hochberg line. Hence, $h(p^*) = h(p)$. □

A.3 Proofs of Section 3.2

Proof of Theorem 8. 1. Let $p \in [0,1]^m$ and $i \in \{1, \ldots, m\}$. It suffices to show that $h_B(p) \supseteq h_B(q)$ for any $q \in [0,1]^m$ given by $q_j = p_j \ \forall j \neq i$ and $q_i > p_i$.

If $p_i > \alpha/m$, then i is non-rejected. Increasing p_i even further will thus not change the result of h_B as $q_i > p_i > \alpha/m$. Therefore, $h_B(q) = h_B(p)$.

If $p_i \leq \alpha/m$ and $q_i > p_i$, the result of $h_B(q)$ depends on whether q_i is greater than α/m or not. In the first case, i is non-rejected in $h_B(q)$, thus $h_B(q) \subset h_B(p)$. In the second case, $q_i \leq \alpha/m$ is still rejected and thus $h_B(q) = h_B(p)$. This proves statement 1.

2. All $q_i \leq \frac{\alpha}{m}$, $i \in h_B(p)$, are rejected, thus $h_B(p) \subseteq h_B(q)$. Similarly, all $q_i > \frac{\alpha}{m}$, $i \notin h_B(p)$, are non-rejected, thus $h_B(p)^c \subseteq h_B(q)^c$. This proves statement 2. □

Proof of Corollary 9. The first statement of Condition 1 is satisfied as h is monotonic by Theorem 8.

Statement 2 of Theorem 8 shows that the Bonferroni correction also satisfies the second statement of Condition 1. Indeed, let $p, q \in [0,1]^m$ be given such that $q_i \leq p_i \ \forall i \in h(p)$ and $q_i \geq p_i \ \forall i \notin h(p)$. By definition of h_B, $p_i \leq \frac{\alpha}{m} \ \forall i \in h_B(p)$ and $p_i > \frac{\alpha}{m} \ \forall i \notin h_B(p)$. In particular, $q_i \leq p_i \leq \frac{\alpha}{m} \ \forall i \in h_B(p)$ and $q_i \geq p_i > \frac{\alpha}{m} \ \forall i \notin h_B(p)$, thus $h_B(p) = h_B(q)$ by statement 2 of Theorem 8. □

Proof of Lemma 10. Let $\delta := \min_{i \in \{1,\ldots,m\}} |p_i^* - \alpha/m|$. Let $p \in [0,1]^m$ with $\|p - p^*\| < \delta$. For all $i \in \{1, \ldots, m\}$, this implies that p_i and p_i^* lie on the same side of the threshold α/m. Therefore, $h_B(p^*) = h_B(p)$. □

A.4 Proofs of Section 3.3

Proof of Lemma 11. Let $i \in h(p, \alpha_1)$. Thus, $\exists j : r_p(j) \geq r_p(i)$ and $m\frac{p_j}{r_p(j)} \leq \alpha_1 \leq \alpha_2$ by definition of h, implying $i \in h(p, \alpha_2)$ and $h(p, \alpha_1) \subseteq h(p, \alpha_2)$. □

Proof of Corollary 12. For $p \leq q$, the estimator $\tilde{\pi}_0$ satisfies

$$\tilde{\pi}_0(p) = \min\left(1, \frac{2}{m}\sum_{i=1}^{m} p_i\right) \leq \min\left(1, \frac{2}{m}\sum_{i=1}^{m} q_i\right) = \tilde{\pi}_0(q).$$

Therefore, as $\frac{\alpha}{\tilde{\pi}_0(p)} \geq \frac{\alpha}{\tilde{\pi}_0(q)}$,

$$h_{PC}(p) = h\left(p, \frac{\alpha}{\tilde{\pi}_0(p)}\right) \supseteq h\left(q, \frac{\alpha}{\tilde{\pi}_0(p)}\right) \supseteq h\left(q, \frac{\alpha}{\tilde{\pi}_0(q)}\right) = h_{PC}(q)$$

by statement 1 of Theorem 5 and Lemma 11. □